基于 Hadoop 的领域术语抽取研究

杜丽萍[1)], 李晓戈[1)], 周元哲[1)], 邵春昌[2)]

[1)](西安邮电大学 计算机学院, 西安 中国 710121)

[2)](中央民族大学 理学院, 北京 中国 100081)

摘 要 传统单机领域术语抽取系统的扩展性已经成为基于大规模语料库进行领域术语抽取的瓶颈。对此提出了一种基于 Hadoop 分布式平台的统计与规则相结合的无监督的专业术语抽取算法，该算法首先利用 PMI（Point-wise Mutual Information）的改进方法确定 2 元待扩展种子，其次采用左右扩展的方式逐字地把 2 元待扩展种子扩展至 2-n 元候选术语（n 表示抽取术语的最大长度，可根据需要指定），最后利用两个基本规则过滤候选术语集合。实验结果表明当 PMI 改进方法的参数取值大于等于 3 时可解决 PMI 方法的缺点、基于大规模语料库进行专业术语抽取的必要性和基于并行算法的高效性。

关键词 术语抽取；专业术语；Hadoop；PMI；PMI 改进方法

中图法分类号 TP391.1

Study of Term Extraction Based on Hadoop

DU Li-Ping[1)], LI Xiao-Ge[1)], ZHOU Yuan-Zhe[1)], SHAO Chun-Chang[2)]

[1)](Department of Computer Science and Technology , Xi'an University of Posts and Telecommunications, Xi'an 710121, China)

[2)](College of Science, Minzu University of China, Beijing 100081, China)

Abstract The expansibility of the traditional stand-alone system has been the main problem and bottleneck for term extraction based on a large-scale corpus. We present a hybrid model for unsupervised identifying terms from the large-scale corpus based on Hadoop, which combines with an improved Point-wise Mutual Information (improved PMI) and some basic rules. Firstly, the method use improved PMI algorithm to determine 2 gram extended seed. Secondly, extending the 2 gram extended seed to 2-n gram candidate term by word-for-word extending to the left or the right(n indicates the maximum length of terms and could be defined as any number as needed). Lastly, the system uses two fundamental rules to filter out candidate terms. The result shows that the improved PMI could solve the problem of low frequency words co-occurrence in PMI method when the parameter value of the improved PMI method is greater than or equal to 3, and also shows that term extraction is necessary based on a large-scale corpus and the system is efficient based on Hadoop.

Key words Term extraction; Technical term; Hadoop; PMI; Improved PMI algorithm

1 引言

专业术语抽取在中文信息处理领域中是一项重要的基础性研究课题。随着科技、经济、文化的快速发展，各个学科领域中的专业术语也发生了很

本课题得到西安邮电大学研究生创新基金 (No. ZL2013-32)资助.杜丽萍，女，1987年生，硕士研究生，主要研究领域为自然语言处理、文本数据挖掘，E-mail: liping.du@qq.com,手机号码：18392385080. 李晓戈，男，1962年生，硕士，教授，主要研究领域为自然语言处理、数据挖掘，E-mail: xiaoge.li@gmail.com. 周元哲，男，1974年生，硕士，讲师，主要研究领域为自然语言处理、机器学习，E-mail: zhouyuanzhe@163.com. 邵春昌，男，1987年生，硕士研究生，主要研究领域为机器学习、数据挖掘，E-mail: haoranqi@yeah.net.

大变化，为了及时了解学科的发展动态，专业术语抽取的需求应运而生。

专业术语抽取方法总体上分为三种:基于规则的方法[1]、基于统计的方法和基于规则与统计相结合的方法。目前，已有很多学者基于不同的方法进行了术语抽取研究。文献[1]利用浅层的语法分析进行名词性专业术语抽取，主要分为分析阶段和解析阶段，在分析阶段利用识别前端标记的基本规则分析文本并且抽取最大长度的名词短语，在解析阶段从最大长度的名词短语中利用语法结构和位置信息抽取可能的术语；文献[2]利用 PMI 方法首先从语料中确定 2 元待扩展种子，其次结合 log-likelihood 方法对 2 元待扩展种子进行扩展，最后根据串频阈值过滤垃圾串；文献[3]首先利用隐马尔科夫模型(HMM, hidden Markov Model)和 KNN(K-Nearest Neighbor)算法进行生物医学实体名称的挖掘，HMM 有效的整合医学领域命名实体的特征，KNN 算法解决了数据稀疏问题；文献[4]从逆文档频数、散串和术语长度 3 个方面对 C-value 方法进行改进得到 IC-value 方法，基于此改进方法进行术语抽取；文献[5]使用子串归并、搭配检测和领域相关度来判断短语结构的完整性、内部搭配的合理性、短语所包含的领域的信息量；文献[6]提出了一种基于规则和统计相结合的方法，首先利用机器学习方法从语料库中获取语言规则，其次利用这些语言规则和停用词表抽取词串列表，接着统计词串的串频和文档频率并根据这些信息过滤，最后衡量词串与领域的相关性；文献[7]采用 PMI 方法和 log-likelihood 方法相结合衡量字串与字串之间“紧密程度”，首先选取二元字串作为待扩展种子，然后再对这些种子进行左右扩展，最后利用再规则过滤；文献[8]首先基于 PMI 方法计算字串内部的“结合强度”，得到术语候选集合，其次过滤掉候选集合中的基本词汇，并利用普通词语搭配前缀、后缀信息对候选集合进一步过滤，最后利用术语的词性构成规则获得最终的抽取结果；文献[9]提出了一种基于混合策略的长术语自动抽取方法，该方法利用 NC-value 参数和 PMI 算法结合识别三字以上的长术语；文献[10]提出使用用户点击的锚文本作为领域术语抽取的语料库，基于串频、左右熵、耦合度、搜索引擎过滤得到候选术语集合，再利用 TFIDF 算法通过背景语料库进行过滤，得到最终的领域术语集合。

上述术语抽取的方法是基于传统单机专业术语抽取算法进行研究的，此类算法适合从较小规模的语料库中识别术语。如今，信息社会已进入大数据（Big Data）时代[11]。数据规模的增长和分布式数据处理技术的发展给人们从大规模语料库中挖掘专业术语带来了巨大机遇[11-13]。因此，在总结前人研究方法的基础上，提出了基于 Hadoop 分布式大数据处理平台的并行术语抽取算法，该方法具有以下优点：

（1）利用 Hadoop 平台解决了大规模数据的可靠存储和计算效率问题，解决了传统单机版算法基于大规模语料库存储困难和运行效率低的缺点；

（2）该方法从未经过分词处理的语料中采用逐字扩展的方式进行术语抽取，避免了有些术语因分词错误导致无法识别的现象，例如“开始集中仓位于兴业银行”的分词结果“开始/v 集中/v 仓/ng 位于/v 兴业/nz 银行/n 。/w”，从这个分词结果中无法识别出术语 “仓位”。

（3）该 PMI 改进算法是通过给 PMI 算法中引进多个字串 x 和字串 y 的联合概率 $p(x,y)$ 因子，解决了 PMI 算法对两个低频且总是一起出现的字串敏感的缺点。

（4）该方法中只利用可存在性过滤和停用词过滤两个基本规则过滤垃圾串，具有很好的可移植性。

2 PMI 方法和 PMI 改进方法

2.1 PMI方法定义及其定理

定义 1. PMI 方法定义如下:

$$PMI(x,y)=\log\frac{p(x,y)}{p(x)p(y)}$$

其中，$p(x)$、$p(y)$ 分别表示字串 x、字串 y 的概率，$p(x,y)$ 表示字串 x 和字串 y 的联合概率，$PMI(x,y)$ 表示字串 x 和字串 y 的相关度，也称 PMI 值。

利用 PMI 方法计算语料库中两个低频且总是相邻出现的字串时，PMI 值要比两个高频字串的 PMI 值高，例如，“锪”和“铢”、“遐”和“迩”、

“鹬”和“蚌”等在语料库中低频且总是相邻出现的字串的 PMI 值高于“我”和“们”等语料库中高频字串的 PMI 值，导致包含这些低频字串的垃圾串的 PMI 值也很高，例如“锱”和“铢必”、“名遐”和“迩”、“鹬”和“蚌相”等，严重的影响了术语抽取的效果。针对此问题，本文采用 PMI^k 方法来解决 PMI 方法的缺点。

抽象语料库中两个低频且总是相邻出现的字串的数学特征和其它字串的数学特征，分别称为低频共现和非低频共现，给出如下定义：

定义 2. 给定字串 str_1、str_2，$p(str_1)$、$p(str_2)$ 分别代表 str_1、str_2 在语料库 *Corpus* 中出现的概率。对于充分小的正数 δ，如果字串 str_1 和 str_2 在 *Corpus* 中总是相邻出现且它们的概率满足 $p(str_1)\le\delta$、$p(str_2)\le\delta$，则称字串 str_1 和 str_2 低频共现；否则，称为非低频共现.

特殊地，低频共现字串 str_1 和 str_2 及它们的组合字串 str_1str_2 的概率满足：

$$p(str_1)=p(str_2)=p(str_1str_2),$$

非低频共现字串 str_1 和 str_2 及它们的组合字串 str_1str_2 的概率满足：

$$\delta<p(str_1str_2)\le p(str_1),$$

$$\delta<p(str_1str_2)\le p(str_2).$$

根据定义 2，给出方法对低频共现字串敏感的定义如下：

定义 3. 设低频共现字串集合：

$$Low=\{(str_1,str_2)\mid \text{字串}str_1\text{和}str_2\text{低频共现}\}$$

非低频共现字串集合：

$$Comm=\{(str_1',str_2')\mid \text{字串}str_1'\text{和}str_2'\text{非低频共现}\}$$

衡量两字串相关度的方法为 f.

如果 $\forall(str_1',str_2')\in Comm$，总存在 $(str_1,str_2)\in Low$，使得通过 f 方法获得的字串 str_1 和 str_2 的相关度总是大于字串 str_1' 和 str_2' 的相关度，则称 f 方法对低频共现字串敏感.

定理 1. PMI 方法对低频共现字串敏感，即对于非低频共现字串集合 *Comm* 中的任意两个字串 c 和 d，总存在低频共现字串集合 *Low* 中两个字串 a 和 b，使得字串 a 和 b 的 PMI 值大于字串 c 和 d 的 PMI 值。

证明. 对于 $\forall(a,b)\in Low$，有

$$p(a)=p(b)=p(a,b),\quad p(a)\le\delta$$

则

$$PMI(a,b)=\log\frac{p(a,b)}{p(a)p(b)}=\log\frac{1}{p(a)}\ge-\log\delta$$

对于 $\forall(c,d)\in Comm$，有

$$\delta<p(c,d)\le p(c),\quad \delta<p(c,d)\le p(d)$$

则

$$PMI(c,d)=\log\frac{p(c,d)}{p(c)p(d)}\le\log\frac{1}{p(c)\ or\ p(d)}<-\log\delta$$

所以，对于 $\forall(c,d)\in Comm$，$\exists(a,b)\in Low$，

$$s.t.\ PMI(a,b)>PMI(c,d).$$

故 PMI 方法对低频共现字串敏感.

证毕.

2.2 PMI改进方法定义及其定理

本文采用的 PMI 改进方法，也称 PMI^k 方法，是通过给 PMI 方法中引进一个或者多个字串 x 与 y 的联合概率因子 $p(x,y)$，来解决 PMI 方法对低频共现敏感的缺点。PMI^k 方法的定义如下：

定义 4. PMI^k 算法定义[14]如下：

$$PMI^k(x,y)=\log\frac{p^k(x,y)}{p(x)p(y)},\quad k\in N^+$$

其中，$p(x)$、$p(y)$分别表示字串 x、y 的概率，$p(x,y)$ 表示字串 x 和 y 的联合概率，$PMI^k(x,y)$ 表示字串 x 和 y 的相关度，也称 PMI^k 值。

特殊地，当 $k=1$ 时，PMI^k 方法即 PMI 方法。

定理 2. 当且仅当正整数 $k\geq 3$ 时，PMI^k 方法解决了对低频共现字串敏感的缺点，即对于低频共现字串集合 *Low* 中任意两个字串 a 和 b，存在非低频共现字串集合 *Comm* 中的两个字串 c 和 d，使得字串 c 和 d 的 PMI^k 值大于字串 a 和 b 的 PMI^k 值。

证明. 分为 $k=1$、$k=2$ 和 $k\geq 3$ 三种情况证明:

（1）当 $k=1$ 时，PMI^k 方法即 PMI 方法，证明如定理 1.

（2）当 $k=2$ 时

对于 $\forall(a,b)\in Low$，有

$$p(a)=p(b)=p(a,b)$$

则

$$PMI^2(a,b)=\log\frac{p^2(a,b)}{p(a)p(b)}=\log 1$$

对于 $\forall(c,d)\in Comm$，有

$$p(c,d)\leq p(c),\quad p(c,d)\leq p(d)$$

则

$$PMI^2(c,d)=\log\frac{p^2(c,d)}{p(c)p(d)}\leq\log 1$$

所以 $k=2$ 时，对于 $\forall(c,d)\in Comm$，存在 $\exists(a,b)\in Low$，$s.t.\ PMI^2(a,b)>PMI^2(c,d)$.

（3）当 $k\geq 3$ 时

对于 $\forall(a,b)\in Low$，有

$$p(a)=p(b)=p(a,b),\quad p(a)\leq\delta$$

则

$$PMI^k(a,b)=\log\frac{p^k(a,b)}{p(a)p(b)}=\log p^{k-2}(a,b)\leq-\log\delta^{k-2}$$

对于 $\forall(c,d)\in Comm$，有

$$\delta<p(c,d)\leq p(c),\quad \delta<p(c,d)\leq p(d)$$

则

$$PMI^k(c,d)=\log\frac{p^k(c,d)}{p(c)p(d)}\geq\log p^k(c,d)$$

令 $p(c,d)=1-\delta$，则 $p^k(c,d)=(1-\delta)^k$

因 $\lim\limits_{\delta\to 0}(1-\delta)^k=1$，$\lim\limits_{\delta\to 0}\delta^{k-2}=0$

所以

$$\exists p(c,d)=1-\delta,\quad s.t.\ PMI^k(a,b)<PMI^k(c,d)$$

故当 $k\geq 3$ 时，对于 $\forall(a,b)\in Low$，$\exists(c,d)\in Comm$，$s.t.\ PMI^k(a,b)<PMI^k(c,d)$.

综上所述，对于 $\forall(a,b)\in Low$，当且仅当 $k\geq 3$ 时，$\exists(c,d)\in Comm$，$s.t.\ PMI^k(a,b)<PMI^k(c,d)$.

故，当且仅当正整数 $k\geq 3$ 时，PMI^k 方法解决了对低频共现字串敏感的缺点.

证毕.

3 基于Hadoop平台的术语抽取系统

术语抽取系统总体上分为三个阶段：第一阶段是大规模语料库的预处理阶段；第二阶段是候选术语抽取阶段，包括 1-n 元字串的串频统计、利用 PMI^k 方法逐字扩展抽取 2-n 元候选术语以及结合基本规则过滤垃圾串；第三阶段是人工判定术语。

3.1 语料库预处理阶段

定义 5. 滑动窗口模型[15-16]. 固定大小的窗口依次从文字序列 $w_1\cdots w_n$ 的开始滑动至结尾，每次滑动一个字的长度，用 W_{ij} 表示当前滑动窗口中所有元素的序列，即文字序列 W 从第 i 个位置到第 j 个位

置的所有序列元素的序列，其中 n 表示文字序列 $w_1 \cdots w_n$ 的长度，$1 \le i \le j \le n$。

假设语料库中文字序列 W 为“#这个办法很给力。$”（‘#’为开始符，‘$’为结束符），滑动窗口大小为 6，利用滑动窗口模型从 W 中依次取得子序列“#这个办法很”、“这个办法很给”、……、“法很给力。$”、“很给力。$”、“给力。$”、“力。$”。图 1 描述了此例在预处理阶段的 Map-Reduce 逻辑数据流图。

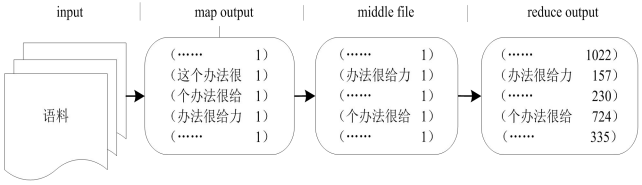

图 1 语料预处理的 Map-Reduce 逻辑数据流图

利用滑动窗口模型进行语料库预处理的优点：（1）只经过一次扫描语料库，得到了所有包含候选术语的字串集合；（2）使统计 1-n 元字串的串频和抽取 2-n 元候选术语的计算不局限于某篇文章内，而是针对整个语料库进行，减小了计算量。

过程 1 描述了利用滑动窗口模型对语料库进行预处理的 Map-Reduce 过程，预处理结果存储于 HDFS 上, 记作 WinData。

过程 1. 预处理阶段的 Map-Reduce 过程：

Map 阶段：

输入：key 值为语料库中文字序列 W 的行号，value 值为 W；

输出：key 值为 W 的子序列 W_{ij}，其中 $1 \le i \le j \le W.length$，value 值为 1；

1) 读取滑动窗口大小 *size* 值;

1）FOR(int $i = 0$; $i < W.length$; $i++$)

2） IF（$i + size < W.length$）

3） THEN $W_{ij} = W.substring(i, i + size)$;

4） ELSE THEN

5） $W_{ij} = W.substring(i, W.length)$;

6） END IF

7） $write(W_{ij}$， 1);

8）END FOR

Reduce 阶段：

输入：key 值为 W_{ij}，其中 $1 \le i \le j \le W.length$，value 值为所有节点上 key 值为 W_{ij} 的 Map 输出的 value 集合〈1，1，…，1〉；

输出：key 值为 W_{ij}，value 值为 W_{ij} 在整个语料库中出现的次数 W_{ij}_count；

1） int $W_{ij}\_count = 0$;

2）FOR(int $i = 0$; $i < value.size()$; $i++$)

3） $W_{ij}\_count++$;

4）END FOR

5）$write(W_{ij}$， $W_{ij}\_count)$;

3.2 统计 1-n 元字串串频

统计语料库中所有的 1-n 元字串的串频，统计结果记为 FreqData，存储在 HBase 上。过程 2 描述了统计串频的 Map-Reduce 阶段。

过程 2. 统计串频的 Map-Reduce 阶段：

Map 阶段：

输入：key 值为 WinData 中子序列 W_{ij} 的行号，value 值为〈W_{ij}， W_{ij}_count〉；

输出：key 值为 W_{ij} 的子序列 W_{im}，其中 $1 \le m \le W_{ij}.length$，value 值为 W_{ij}_count；

1） FOR (int $m = 1$; $m \le W_{ij}.length$; $m++$)

2） $W_{im} = W_{ij}.substring(0, m)$;

3） $write(W_{im}$， $W_{ij}\_count)$;

4） END FOR

Reduce 阶段：

输入：key 值为W_{im}，value 值为所有节点上 key 值为W_{im}的 Map 输出的 value 集合<W_{ij}_count，W'_{ij}_count，…>；

输出：key 值为W_{im}，value 值为W_{im}在整个语料库中出现的次数W_{im}_count；

1) int $W_{im}\_count = 0$;

2) FOR(int $i = 0$; $i < value.size()$; $i++$)

3) $W_{im}\_count++$;

4) END FOR

5) $write(W_{im}, W_{im}\_count)$

3.3 术语抽取

图 2 描述了抽取术语算法的流程图，该算法首先抽取 2 元待扩展种子，其次将 2 元待扩展种子左右扩展至 2-n 元候选术语，最后利用可存在性过滤规则和停用词过滤规则实施过滤。

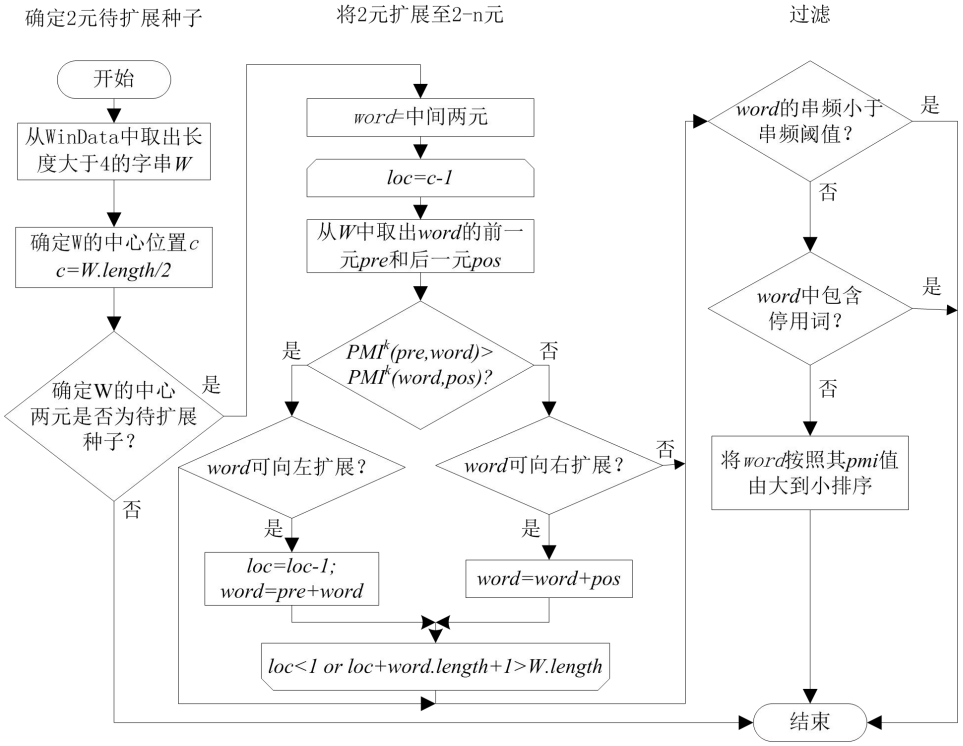

图 2 术语抽取算法流程图

3.3.1 确定 2 元待扩展种子

算法 1 描述了 2 元待扩展种子的抽取算法。从 WinData 中取出长度大于 4 的子序列W_{ij}，确定W_{ij}的中心位置c，取出W_{ij}中间 4 元字串$W_{c-2}W_{c-1}W_cW_{c+1}$，分别计算$W_{c-2}W_{c-1}$、$W_{c-1}W_c$、W_cW_{c+1}的 PMI^{k}值，令

$$mean_1 = 1/2\left(PMI^k\left(W_{c-2},W_{c-1}\right)+PMI^k\left(W_{c-1},W_c\right)\right)$$

，

$$mean_2 = 1/2\left(PMI^k\left(W_{c-1},W_c\right)+PMI^k\left(W_c,W_{c+1}\right)\right)$$

如果 $PMI^k\left(W_{c-1},W_c\right) > PMI^k\left(W_{c-2},W_{c-1}\right)+mean_1$ 并且 $PMI^k\left(W_{c-1},W_c\right) > PMI^k\left(W_c,W_{c+1}\right)+mean_2$，则认为字串$W_{c-1}W_c$是一个词或者词的一部分的可能性很大，将字串$W_{c-1}W_c$确定为待扩展种子；否则，

认为字串 W_{c-1}、W_c 各自成词或是词的边界，将 FreqData 中字串 $W_{c-1}W_c$ 的串频减 1，退出。

算法 1. 2 元待扩展种子抽取算法

输入：WinData 中长度大于 4 的子序列 W_{ij}；

输出：2 元待扩展种子 $W_{c-1}W_c$ 或不输出任何值；

1） $c = W_{ij}.length/2$；

2） $W_{c-2}W_{c-1}W_cW_{c+1} = W_{ij}.substring(c-2, c+2)$；

3）从 FreqData 中依次取出 W_i、W_jW_{j+1} 的串频，其中 $c-2 \le i \le c+1$ 且 $c-2 \le j \le c$；

4）分别计算字串 W_j、W_{j+1} 的 PMIk 值，其中 $c-2 \le j \le c$；

5）计算 $mean_1$ 和 $mean_2$

$mean_1 = 1/2\left(PMI^k(W_{c-2}, W_{c-1}) + PMI^k(W_{c-1}, W_c)\right)$

$mean_2 = 1/2\left(PMI^k(W_{c-1}, W_c) + PMI^k(W_c, W_{c+1})\right)$；

6) IF($PMI^k(W_{c-1}, W_c) > PMI^k(W_{c-2}, W_{c-1}) + mean_1$

&& $PMI^k(W_{c-1}, W_c) > PMI^k(W_c, W_{c+1}) + mean_2$)

7)THEN $W_{c-1}W_c$ 为待扩展种子;

8)ELSE THEN

9) 将 FreqData 中 $W_{c-1}W_c$ 的串频键 1;

10) 退出;

11)END IF

3.3.2 扩展 2 元待扩展种子至 2-n 元候选术语

算法 2 描述了把 2 元待扩展种子扩展至 2-n 元候选术语的算法。对于给定子序列 W_{ij}，如果中间两元 $W_{c-1}W_c$ 是待扩展种子，则令 $word = W_{c-1}W_c$，$loc = c-1$（*loc* 记录 *word* 的第一个字在 W_{ij} 中的位置），$pmi = PMI^k(W_{c-1}, W_c)$，循环的对 *word* 进行左右扩展：从 W_{ij} 中取出 *word* 的前一元字串 *pre* 和后一元字串 *pos*，分别计算 *pre*、*word* 和 *word*、*pos* 的 PMIk 值，分为两种情况：

(1) 如果 $PMI^k(pre, word) > PMI^k(word, pos)$，则比较 *pre*、*word* 的 PMIk 值和 *pmi*，令 $mean = 1/2\left(PMI^k(pre, word) + pmi\right)$，如果 $PMI^k(pre, word) + mean > pmi$，则认为 *word* 可向前扩展，令 $word = pre + word$，$loc = loc - 1$，$pmi = PMI^k(per, word)$，继续左右扩展，否则输出 *word* 及对应的 *pmi*，退出扩展。

（1）如果 $PMI^k(pre, word) \le PMI^k(word, pos)$，则比较 *word*、*pos* 的 PMIk 值和 *pmi*，令 $mean = 1/2\left(PMI^k(word, pos) + pmi\right)$，如果 $PMI^k(word, pos) + mean > pmi$，则认为 *word* 可向后扩展，令 $word = word + pos$，$pmi = PMI^k(word, pos)$，继续左右扩展，否则输出 *word* 及对应的 *pmi*，退出扩展。

算法 2. 扩展 2 元待扩展种子至 2-n 元候选术语算法

输入：WinData 中子序列 W_{ij} 及它的中心两元 $W_{c-1}W_c$；

输出：2-n 元候选术语 *word* 及对应的 *pmi*；

1) WHILE($loc - 1 \ge 0$ &&

$loc + word.length < W_{ij}.length$)

2) $word = W_{c-1}W_c$；

3) $loc = c-1$；

4) $pmi = PMI^k\left(W_{c-1}, W_c\right)$；

5) $temp = loc + word.length$；

6) $pre = W_{ij}.substring\left(loc-1, loc\right)$；

7) $pos = W_{ij}.substring\left(temp, temp+1\right)$；

8) 分别计算 *pre*、*word* 和 *word*、*pos* 的 PMI^k 值；

7) IF($PMI^k\left(pre, word\right) > PMI^k\left(word, pos\right)$)

8) THEN

$mean = 1/2\left(PMI^k\left(pre, word\right) + pmi\right)$；

9) IF($PMI^k\left(pre, word\right) + mean > pmi$)

10) THEN $word = pre + word$；

11) $loc = loc - 1$；

12) $pmi = PMI^k\left(per, word\right)$；

13) ELSE THEN

输出 *word* 及对应的 *pmi*；

14) 退出扩展；

15) END IF

16) ELSE THEN

$mean = 1/2\left(PMI^k\left(word, pos\right) + pmi\right)$；

17) IF($PMI^k\left(word, pos\right) + mean > pmi$)

18) THEN $word = word + pos$；

19) $pmi = PMI^k\left(word, pos\right)$；

20) ELSE THEN

输出 *word* 及对应的 *pmi*；

21) 退出扩展；

22) END IF

23) END IF

24)END WHILE

3.3.3 规则过滤

可存在性过滤规则和停用词过滤规则的定义如下：

定义 6. 可存在性过滤规则. 给定串频阈值 *Threshold* 和字串集合 *Cands*. 对于任意字串 $str \in Cands$，如果字串 *str* 的串频 $Freq\left(str\right)$ 满足 $Freq\left(str\right) \le Threshold$，那么从字串集合 *Cands* 中删除字串 *str*.

定义 7. 停用词过滤规则. 给定字串 $w_1w_2\cdots w_m$ 和停用词集合 *Filter*，对于任意的字串 $w_1'w_2'\cdots w_l' \in Filter$，如果 $\exists j \in \left[0, m-l\right]$，$s.t.\ w_i' = w_{i+j}$，$i = 1,2,\cdots,l$，那么删除字串 $w_1w_2\cdots w_m$，其中，$m, l \in N^+$，$l \le m$.

利用可存在性过滤规则实施过滤的方法：如果候选术语 *word* 在 FreqData 中的串频小于串频阈值 *Threshold*，则将 *word* 过滤掉，退出；否则保留 *word*，进行下一步过滤。

利用停用词表过滤规则实施过滤的方法：如果候选术语 *word* 中包含字串 $w_1'w_2'\cdots w_l' \in Filter$，则将 *word* 过滤掉，退出；否则输出 *word* 及其对应的 *pmi*，结束。

过程 3 描述了整个术语抽取阶段 Map-Reduce 过程，Map 阶段利用算法 1 和算法 2 抽取出 2-n 元候选术语，在 Reduce 阶段，运用可存在性过滤规则和停用词过滤规则对候选术语进行过滤。抽取的术语存储于 HDFS 上，记作 Terms.

过程 3. 术语抽取阶段的 Map-Reduce 过程

setup 函数：

读取串频阈值 *Threshold*；

读取停用词集合 *Filter* 中的所有字串，存放在内存 *FilterSet* 中；

Map 阶段：

输入：key 值为 WinData 中子序列 W_{ij} 的行号，value 值为$\langle W_{ij}$，W_{ij} 对应的 $W_{ij}\_count\rangle$；

输出：key 值为候选术语 *word*，value 值为 *word* 的 PMI^k 值 *pmi*；

1)利用算法 1 确定 2 元待扩展种子；

2)利用算法 2 把 2 元待扩展种子扩展至 2-n 元候选术语；

Reduce 阶段：

输入：key 值为候选术语 *word*，value 值为 *word* 的 PMI^k 值 *pmi*；

输出：key 值为候选术语 *word*，value 值为 *word* 的 PMI^k 值 *pmi*；

1)从 FreqData 中读取 *word* 的串频 *word _ freq*；

2) IF($word\_freq \le Threshold$)

3) THEN 退出；

4) ELSE THEN

5) FOR(String w : *Filter*)

6) IF($word.contains(w)$)

7) THEN 退出；

8) END IF

9) END FOR

10) END IF

11) *write*(*word*， *pmi*)；

3.4 术语判定

术语判定分为两个步骤：

（1）给定核心词表 *Core*，对于 $\forall word \in Terms$，如果 $word \in Core$，将 *word* 添加到核心词汇集合 *Core'* 中；否则，按 *word* 的 PMI^k 值 *pmi* 降序的把 *word* 添加到术语集合 *Terms'* 中。

（2）对 *Terms'* 中每个词 *word* 进行人工判定，如果 *word* 不是正确的术语，则将 *word* 添加到垃圾串集合 *Garbs'* 中；否则将 *word* 添加到术语集合 *Corrects* 中。

4 实验结果与分析

4.1 实验数据

(1) 1G 财经领域语料，用于抽取财经领域的术语；

(2) 200M 新闻领域语料，用于抽取新闻领域的术语；

(3) 停用词典：包含 702 个停用词，用于过滤候选术语集合中的垃圾串；

(4) ICTCLAS 核心词典：共收集了 79 836 个词语，是目前比较规范的词典之一，用于判定专业术语。

4.2 实验结果

由于难以获得大规模语料库中所有出现的术语，所以不能使用召回率评测术语抽取结果，只采用准确率来衡量实验结果的好坏。准确率的计算公式分别如下：

$$准确率=\frac{正确术语条数}{术语条数}\times 100\%.$$

针对 1G 财经领域语料库实验，给定串频阈值 *Threshold*=6，在 PMI^k 方法的参数 k 分别取 1 到 10 之间 10 个正整数值时进行术语抽取实验。由于 PMI^k 方法的参数 k 值不同，抽取的候选术语条数也不同，为了有效对比不同 k 值下的抽取结果，本文只选取候选术语的前 2000 条结果做评测，表 1 描述了财经领域的实验结果，其中，候选术语指经过术语抽取系统抽取的结果条数；核心词汇指候选术语中包含在 ICTCLAS 核心词典中的词汇条数；术语指从候选术语中剔除掉包含在 ICTCLAS 核心词典中的词汇后剩余的词汇条数；正确率分别是前 500、前 1000、前 1500、前 2000 条术语的正确率。

表 1 1G 财经领域实验结果

k 值	候选术语	核心词汇	术语	正确率(%)			
				500	1000	1500	2000
1	269460	25812	243648	31.2	25.5	22.53	21.2
2	133908	26261	107647	31.8	28.2	27.13	27.35
3	61248	25537	35711	63.4	56.1	51.53	48.95
4	41061	23241	17820	75	67.6	62.33	59.5
5	30138	20052	10086	81.2	74.7	69.53	67.05
6	23609	17129	6480	85.6	78.9	76	74.05
7	19355	14822	4533	91.2	84.2	83	81.85
8	16616	13167	3449	94	90.3	88.67	88.95
9	14693	11885	2808	94.6	91.99	90.47	91.3
10	13442	11025	2417	95.2	92.4	91.6	92.6

表 2 列举了 PMI^k 方法的参数 k 分别取 1 到 10 之间 10 个正整数值时术语抽取结果的前 20 条。

正如表 2 所示，$k=1$、$k=2$ 时结果中排在前面的大多数为 3 元以上的字串，当 $3\le k\le 10$ 时，结果中排在前面的大多数为 2 元字串，并且随着 k 的增大结果中 2 元字串的比列也增大，在参数 $3\le k\le 10$ 时，结果中的 2 元字串均占到 70%以上。

表 2 财经领域前 20 条实验结果

$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	$k=10$
荽愈肝	片仔癀	升浪	升浪	升浪	升浪	升浪	升浪	升浪	升浪
倜傥	道扬镳	蓝筹	主升	主升	主升	主升	主升	主升	主升
镅铢必	简意赅	散户	博客	点击	点击	点击	点击	点击	点击
喹诺酮	火如荼	博客	点击	短线	短线	短线	短线	短线	短线
名遐迩	饕餮	主升	短线	博客	博客	博客	博客	博客	选股
耄耋	偃旗息	鼎砥	散户	权重	权重	权重	权重	选股	博客
马齐喑	何鸿燊	私募	蓝筹	散户	选股	选股	选股	权重	权重
虎作伥	草甘膦	狼嗥	权重	蓝筹	散户	操盘	操盘	操盘	操盘
如弊屣	沆瀣	短线	涨停	涨停	居前	涨停	涨停	涨停	涨停
酗灌顶	贵研铂	点击	私募	居前	涨停	散户	均线	均线	均线
血化瘀	驰宏锌锗	权重	居前	选股	操盘	均线	散户	散户	散户
吴昌铣	方金钰	涨停	鼎砥	操盘	均线	蓝筹	博文	博文	博文
桀骜	卫文麾	居前	狼嗥	均线	蓝筹	博文	较多	较多	较多
捶胸顿	吴敬琏	飘红	选股	博文	博文	较多	蓝筹	蓝筹	蓝筹
選擇	殚精竭	玉名	均线	私募	较多	私募	股基	股基	股基
黾农老	辕北辙	翻红	操盘	鼎砥	私募	两市	两市	两市	两市
堂吉诃	龚方雄	均线	博文	狼嗥	狼嗥	股基	私募	日线	日线
阚治东	醌酮	菲特	净流	较多	鼎砥	缩量	日线	缩量	缩量
殚精竭	囵吞枣	威廉	力动	玉名	缩量	日线	缩量	私募	放量
繼續	州固锝	抄底	玉名	翻红	减仓	减仓	放量	放量	私募

针对 200M 新闻领域语料库实验,在和 1G 财经领域语料库相同的条件下实验，表 3 描述了实验结果，其中，候选术语、核心词汇、术语的意义同表 1。

表 3 200M 新闻领域实验结果

k 值	候选术语	核心词汇	术语
1	30458	9606	20825
2	20279	9538	10741
3	11171	8714	2457
4	7531	6590	941
5	5309	4858	451
6	3955	3699	256
7	3187	3022	165
8	2697	2587	119
9	2326	2239	87
10	2073	2004	69

图 3 描述了在 1G 的语料库规模下，该术语抽取系统的加速比随着 Hadoop 平台节点个数的变化趋势。

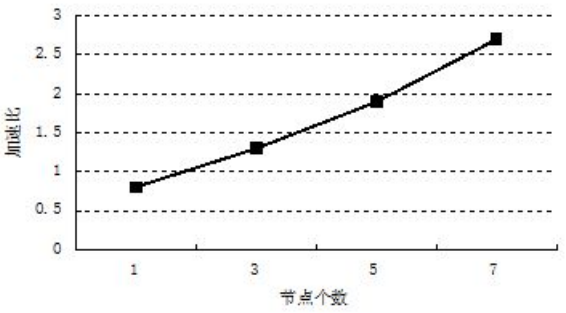

图 3 加速比与节点个数关系图

4.3 结果分析

由表 1 可看出以下几点：

(1) 随着 PMI^k 方法参数 k 取值增大，抽取的候选术语条数、包含在 ICTCLAS 核心词典中的词汇条数、术语条数均逐渐减少，但术语的前 500、前 1000、前 1500、前 2000 条的准确率均随着 k 值增大而增大。

(2) 当 $k=1$、$k=2$ 时，实验分别抽取出 243648 条、107647 条术语，远远大于一个常规词典所包含的术语条数，说明该结果中包含了大量的垃圾串。

(3) 当 $k=3$ 时前 2000 条术语的准确率约是 $k=1$、$k=2$ 时的两倍，说明当 PMI^k 方法参数 k 取值大于等于 3 时明显改善了 PMI 方法的效果。

从表 2 可看出，当 PMI^k 算法的参数 k 取值为 1、2 时和参数 k 取值为 3 到 10 之间的正整数时术语结果差异较大。在参数 k 取值为 1、2 的结果中，排名在前的字串中均包含低频的字或词，例如垃圾串“镅铢必”、“名遐迩”、“酗灌顶”、“道扬镳”中分别包含“镅铢”、“遐迩”、“酗”、“镳”等低频字串且这些字串的搭配词语固定，该现象反映出利用 PMI 算法、PMI^2 算法对低频共现字串敏感的缺点；在参数 k 取值为 3 到 10 之间的正整数的结果中，结果中均没有出现低频共现字串，说明参数 $3\le k\le 10$ 时 PMI^k 算法克服了对低频共现字串敏感的缺点，并且从结果可以看出 k 取值为 3 到 10 之间的正整数时结果非常相似：“升浪”、“点击”、“短线”等字串随着 k 值的变化排名几乎不变，说明参数 $3\le k\le 10$ 时 PMI^k 方法对相关性高的字串收敛。该结果证实了定理 1 和定理 2，即 PMI 方法存在对低频字串敏感的缺点，当 PMI^k 方法参数 $k\ge 3$ 时可以解决 PMI 方法对低频共现字串敏感的缺点，与理论证明一致。

表 3 中，随着 PMI^k 方法参数 k 取值增大，抽取的候选术语条数、包含在 ICTCLAS 核心词典中的词汇条数、术语条数均逐渐减少，同时，由于语料规模较小，当 $3\le k\le 10$ 时抽取的术语条数均较少；统计参数 k 值取 3、4 时前 500 条结果的准确率，分别为 58.4%和 69%，对比 1G 语料库规模下参数 k 值取 3、4 时前 500 条结果的准确率 63.4%、75%，可以看出，基于大规模语料库可提高术语抽取结果的准确率；所以基于大规模语料进行术语抽取是必要的。

由图 3 可以看出，在 1G 的语料库规模下，Hadoop 平台的节点个数在一定范围内，加速比随着节点个数的增加趋于直线增长，说明基于分布式平台进行术语抽取可有效减少时间开销。

5 结论

本文提出了一种基于 Hadoop 平台的利用 PMI^k 方法进行无监督地抽取术语的算法，实验结果表明当 PMI^k 方法的参数 k 值大于等于 3 时可解决 PMI 方法的缺点，与理论证明相一致。本文还验证了基于大规模语料库进行术语抽取的必要性和基于分布式平台进行术语抽取的高效性。利用该术语抽取方法，在 400M 百度贴吧语料库中进行新词发现实验，实验结果在 $k=5$ 时抽取出 976 条新词，其中 843 条是正确新词，准确率达到 86.37%，结果说明该术语抽取算法具有可移植性。文中人工的选取 PMI^k 方法的参数 k 值进行了实验，下一步工作是根据应用要求达到的准确率、语料库规模以及语料体裁等因素自适应地确定参数 k 的值，并且优化判别可扩展的条件，提高系统抽取长术语的能力。

参 考 文 献

[1] Bourigault D. Surface Grammatical Analysis for the Extraction of Terminological Noun Phrases[C] M Proceedings of COLING. 92. 1992: 977- 981.

[2] Pantel P, Lin D. A Statistical Corpora-based Term Extractor[C] M Lecture Notes in Artificial Intelligence. Springer, Verlag, 2001: 34- 46.

[3] Zhou, G., et al. Recognizing names in biomedical texts: a machine learning approach[J] . Bioinformatics. 2004: 78- 90.

[4] Hu A-Pei, Zhang Jing, Liu Jun-Li. Chinese term extraction based on improved C-value method. New Technology of Library and Information Service, 2013(2): 24-29 (in Chinese)
(胡阿沛, 张静, 刘俊丽. 基于改进C-value方法的中文术语抽取. 现代图书情报技术, 2013(2): 24-29)

[5] Zhou Lang, Shi Shu-Min, Feng Chong, Huang He-Yan. A Chinese term extraction system based on multi-strategies integration. Journal of China Society for Scientific and Technical Information, 2010, 29(3): 460-467 (in Chinese)
(周浪, 史树敏, 冯冲, 黄河燕. 基于多策略融合的中文术语抽取方法. 情报学报, 2010, 29(3): 460-467)

[6] Zhou Lang, Zhang Liang, Feng Chong, Huang He-Yan. Terminology extraction based on statistical word frequency distribution variety. Computer Science, 2009, 36(5): 177-180 (in Chinese)
(周浪, 张亮, 冯冲, 黄河燕. 基于词频分布变化统计的术语抽取方法. 计算机科学, 2009, 36(5): 177-180)

[7] Du Bo, Tian Huai-Feng, Wang Li, Lu Ru-Zhan. Design of domain –specific term extractor based on multi-strategy. Computer Engineering, 2005, 31(14): 159-160 (in Chinese)
(杜波, 田怀凤, 王立, 陆汝占. 基于多策略的专业领域术语抽取器的设计. 计算机工程, 2005, 31(14): 159-160)

[8] Zhang Feng, Xu Yun., Hou Yan, Fan Xiao-Zhong. Chinese term extraction system based on mutual information. Application Research of Computers, 2005, 22(5): 72-74 (in Chinese)
(张峰, 许云, 候艳, 樊孝忠. 基于互信息的中文术语抽取系统. 计算机应用研究, 2005, 2005, 22(5): 72-74)

[9] Liang Ying-Hong, Zhang Wen-Jing, Zhou De-Fu. A hybrid strategy for high precision long term extraction. Journal of Chinese Information Processing, 2009, 23(6): 26-30 (in Chinese)

(梁颖红, 张文静, 周德福. 基于混合策略的高精度长术语自动抽取. 中文信息学报, 2009, 23(6): 26-30)

[10] Yan Xing-Long, Liu Yi-Qun, Fang Qi, Zhang Min, Ma Shao-Ping, Ru Li-Yun. Domain-specific terms extraction based on Web resource and user behavior. Journal of Software, 2013, 24(9): 2089-2100 (in Chinese)
(闫兴龙, 刘奕群, 方奇, 张敏, 马少平, 茹立云. 基于网络资源与用户行为信息的领域术语提取. 软件学报, 2013, 24(9): 2089-2100)

[11] Wang Yuan-Zhou, Jin Xiao-Long, Cheng Xue-Qi. Network big data: present and future. Chinese Journal of Computers, 2013, 36(6): 1125-1138 (in Chinese)
(王元卓, 靳小龙, 程学旗. 网络大数据：现状与展望. 计算机学报, 2013, 36(6): 1125-1138)

[12] Wang S, Wang HJ, Qin XP, Zhou X. Architecting big data: Challenges, studies and forecasts. Chinese Journal of Computers, 2011,34(10):1741−1752 (in Chinese with English abstract). [doi: 10.3724/SP.J.1016.2011 01741].

[13] Meng XF, Ci X. Big data management: Concepts, techniques and challenges. Journal of Computer Research and Development, 2013, 50(1):146−169 (in Chinese with English abstract).

[14] G. Bouma. Normalized(point-wise) mutual information in collocation extraction[J]. In Proc. Boennial GSCL Conference 2009, Meaning: Processing Texts Automatically, pp. 31-40, Tbingen, Gunter Narr Verlag, (2009).

[15] Li Jian-Zhong, Zhang Dong-Dong. Algorithms for dynamically adjusting the sizes of sliding windows. Journal of Software, 2004, 15(12): 1800-1813 (in Chinese)
(李建中, 张冬冬. 滑动窗口规模的动态调整算法. 软件学报, 2004, 15(12): 1800-1813)

[16] Wang Xu, Li Jian-Zhong, Wang Wei-Ping. Processing compressed sliding window continuous queries over data streams. Journal of Computer Research and Development, 2004, 41(10): 1639-1644 (in Chinese)
(王栩, 李建中, 王伟平. 基于滑动窗口的数据流压缩技术及连续查询处理方法. 计算机研究与发展, 2004, 41(10): 1639-1644)

DU Li-Ping, born in 1987, M. S. candidate. Her current research interests include nature language processing and text data mining.

LI Xiao-Ge, born in 1962, M. S., professor. His current research interests include nature language processing and data mining.

ZHOU Yuan-Zhe, born in 1974, M. S., assistant professor. His current research interests include nature language processing and machine learning.

SHAO Chun-Chang, born in 1987, M. S. candidate. His current research interests include machine learning and data mining.

Background

Term extraction is a basic and important research topic in the Chinese information processing filed. The main purpose of term extraction is used to extract features of a professional field, unsupervised establish professional dictionary, and chase the dynamic and changing of a field.

A large amount of work has been done in term extraction. Basically, there are three kinds of assets in term extraction: rule-based, statistical-based, and based on the combination of rules and statistics. In previously term extraction methods, researchers tend to use pure rule-based or statistical-based method, but unfortunately the two methods have their own shortcomings, the problem of rule-based method is the poor translation in different fields and the disadvantage of statistical-based approach is garbage string filtering problem. Nowadays, most researchers might like using the combination of rules and statistics, they might compute the correlation degree using statistical-based method and then using rule-based to filter out the garbage in results. At present, these methods are based on the traditional stand-alone system and the results is not prrety.

With the rapid development of the Web2.0, the age of big data into every aspect of our lives. Big data and distributed processing technology provide the great opportunities to processing, analysis and data mining for us. In this paper, we combine big data and hadoop platform to extract term in a corpus. We investigate on the weak point of PMI method under the circumstance of computing two low frequency and co-occurrence words, and then we abstract the mathematical features of two low frequency and co-occurrence words in corpus and we prove the improved PMI method, PMI^k method could solve the problem of PMI method when the parameter value of PMI^k method is greater than or equal to 3 in both theory and practice. Meanwhile, the result shows that term

extraction is necessary based on a large-scale corpus and our system is efficient based on Hadoop.

The study results of this thesis will be used to update professional dictionary and thus improve the precision of the existing word segmentation system.